**PRACTICAL - 4**

**Aim: Execute the value matching and pattern matching conditions on the bank’s schema to satisfy the following requirements:**

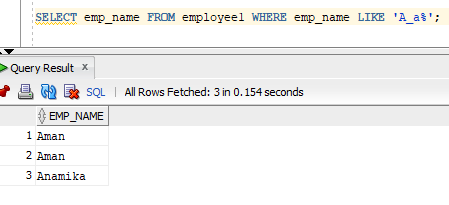
**Theoretical Description:**

* Used to **retrieve data** from the database without modifying it. The primary command is **SELECT**, which allows querying data based on conditions.
* The **LIKE** operator is used for **pattern matching** in string values. It allows us to search for data that matches a particular pattern, such as certain characters at specific positions in a string.
* **Value Matching**: Used to find exact matches or ranges in values (e.g., checking branch names or salary values).
* **Pattern Matching**: Used to find data based on patterns in strings (e.g., employee names, job IDs).
* Each task uses the SELECT statement with **LIKE** for pattern matching and simple operators for value matching. This allows filtering employees, customers, and jobs based on the given conditions.
* The ESCAPE keyword allows treating special characters (like \_ and %) as literal characters in pattern matching. For example, '\\\_%' will search for an underscore (\_) in the string.

**Query-1:Display all employee whose name start with ‘A’ and third character is ‘a’.**

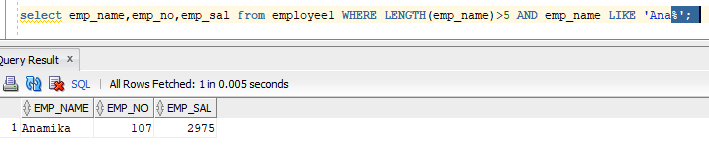
**SELECT emp\_name FROM employee1 WHERE emp\_name LIKE 'A\_a%';**

**Output:**

****

**Query-2: Display name, number and salary of those employees whose name is 5 characters long and first three characters are ‘Ana’.**

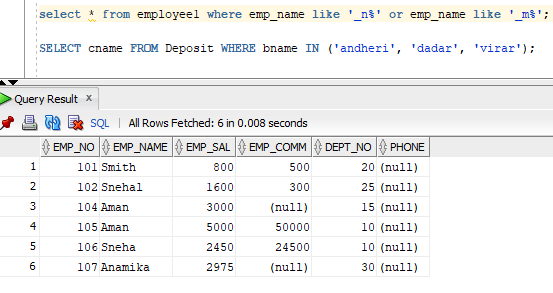
**select emp\_name,emp\_no,emp\_sal from employee1 WHERE LENGTH(emp\_name)>5 AND emp\_name LIKE 'Ana%';**

**Output: **

**Query-3: Display all information of employee whose second character of name is either ‘M’ or ‘N’.**

**SQL Statement: select \* from employee1 where emp\_name like '\_n%' or emp\_name like '\_m%';**

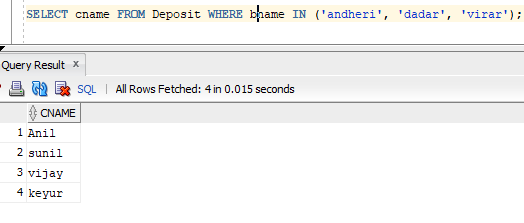
**Output:**

****

**Query-4: Find the list of all customer name whose branch is in ‘andheri’ or ‘dadar’ or ‘virar’.**

**SQL Statement: SELECT cname FROM Deposit WHERE bname IN ('andheri', 'dadar', 'virar');**

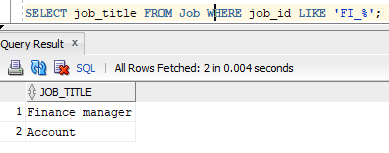
**Output:**

****

**Query-5: Display the job name whose first three character in job id field is ‘FI\_’.**

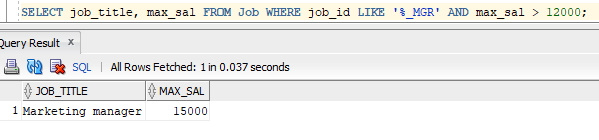
**SQL Statement: SELECT job\_title FROM Job WHERE job\_id LIKE 'FI\_%';**

**Output:**

****

**Query-6: Display the title/name of job whose last three character are ‘\_MGR’ and there maximum salary is greater than Rs 12000.**

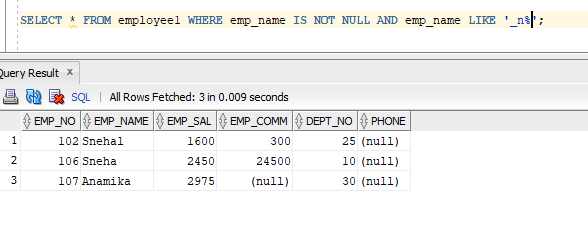
**SQL Statement: SELECT job\_title, max\_sal FROM Job WHERE job\_id LIKE '%\_MGR' AND max\_sal > 12000;**

**Output:   
**

**Query-7: Display the non-null values of employees and also employee name second character should be ‘n’ and string should be 5 character long.**

**SQL Statement: SELECT \* FROM employee1 WHERE emp\_name IS NOT NULL AND emp\_name LIKE '\_n%';**

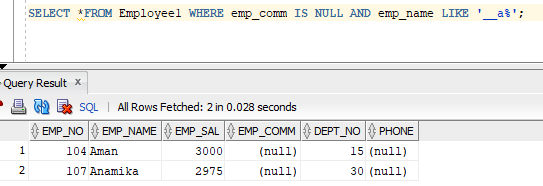
**Output:**

****

**Query-8: Display the null values of employee and also employee name’s third character should be ‘a’.**

**SQL Statement: SELECT \*FROM Employee1 WHERE emp\_comm IS NULL AND emp\_name LIKE '\_\_a%';**

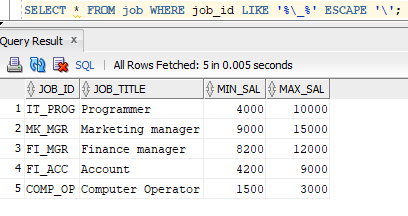
**Output:**

****

**Query-9: Display employee no, name and department details of those employee whose department no is between 15 and 25.**

**SQL Statement: SELECT \* FROM job WHERE job\_id LIKE '%\\_%' ESCAPE '/';**

**Output:**

****

**Question-1: how to show first five row of table?**

**Answer:** SQL Statement : SELECT \* FROM table\_name WHERE ROWNUM <= 5;

**Question-2: how to delete duplicate records?**

**Answer**: SQL Statement : DELETE FROM table\_name WHERE ROWID NOT IN (SELECT MIN(ROWID) FROM table\_name GROUP BY column\_name);

**Question-3:difference between delete , drop and transcate.**

**Answer:**

* **DELETE** → Removes selected rows (WHERE can be used), logs changes, can be rolled back.
* **DROP** → Removes the table completely (structure + data), **cannot** be rolled back.
* **TRUNCATE** → Removes all rows quickly, keeps structure, **cannot** be rolled back in most databases.